

Research Article

A Two-Stage Train Delay Prediction Method Based on Data Smoothing and Multimodel Fusion Using Asymmetry Features in Urban Rail Systems

Hao Wang 

China Railway Siyuan Survey and Designing Group Co., LTD, Wuhan, China

Correspondence should be addressed to Hao Wang; 1005300064@qq.com

Received 3 December 2021; Revised 12 May 2022; Accepted 24 May 2022; Published 22 June 2022

Academic Editor: Antonio De Domenico

Copyright © 2022 Hao Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

During the operation of urban rail transit trains, train delays have a great negative impact on subway safety and operation management. It is caused by various external environmental disturbances or internal equipment failures. In order to better grasp the situation of train delays, adjust train operation schedules in time, and improve the quality of rail transit command and transportation services, this paper proposes a two-stage train delay prediction method based on data smoothing and multimodel fusion. In the first stage, the Singular Spectrum Analysis (SSA) method is used to smooth the train operation data, and the smoothed components and residual components are extracted. In the second stage, we use different machine learning methods to train the smoothed data and use the K -nearest neighbor (KNN) method to fuse different trainers. Finally, the predicted values of the smoothed and residual components are combined to improve the overall performance of the train delay prediction model, especially under asymmetry features. The research results show that the prediction accuracy of the two-stage train delay prediction method based on KNN multimodel fusion is significantly better than that of the independent machine learning model.

1. Introduction

Urban rail transit is currently an internationally recognized green transportation with low energy consumption and large capacity. With its characteristics of fast speed, high efficiency, and low pollution, it has become one of the main travel modes to alleviate urban traffic pressure. As of the end of 2020, 45 cities in China have opened 244 urban rail transit operating lines with a total length of 7969.7 kilometers. According to the “Outline for Building a Powerful Transportation Country” put forward in September 2019, the Chinese transportation development goal is “one-hour commute in metropolitan areas, two-hour access to urban agglomerations, and three-hour coverage of major cities across the country.” Therefore, the transportation mode dominated by urban rail public transportation is an inevita-

ble trend of development [1]. With the continuous increase of subway lines, operating length, and expansion of passenger traffic, the composition of the technical system of urban rail transit organization will become more complex, and the business systems responsible for linkage will increase with the informatization of rail transit. The future urban rail transit safety guarantee system will face new opportunities and challenges. For the scientificity and reliability of train operation safety management, real-time monitoring and control of train operation status are of great significance to the safety of urban rail transit organizations [2, 3].

The subway is a complex system, and its operation process is often affected by factors such as passenger flow changes and equipment failures. For example, during the morning and evening rush hours, the stop time is prolonged due to passenger flow congestion, and the interval running

time is too long due to improper driving of the driver or equipment failure. This in turn causes the train to be delayed at the next station [4]. Under high-density traffic conditions, once a train is delayed in urban rail transit, it is easy to cause subsequent delays, spread to other trains on the line, and disrupt the order of train operation. Therefore, grasping the situation of train delay is one of the core tasks of train dispatching, and realising the prediction of train delay is of great significance for improving the quality of train dispatching and command work [5]. However, in the current on-site dispatching operations, the train delay prediction is often after a certain section is delayed. The train dispatcher makes manual judgments on the extent and scope of the train delay based on his own experience and then adjusts the operation diagram [6]. This kind of delay prediction that relies on personnel experience cannot make timely and accurate quantitative judgments about train delays. Therefore, it is necessary to improve the real-time prediction ability of train delay time. Accurately predicting the train delay time enables the dispatcher to estimate the operating status of the train more accurately and formulate driving command decisions more reasonably and has certain practical significance in improving the quality of rail transit driving command [7, 8].

Many scholars have conducted in-depth research on the problem of train delays. Harrod et al. [9] realised the prediction of the total delay time of the train on a certain line by studying the buffer time and the running time of the late train by using the closed analytic formula based on the theory of delay propagation and verified it by experiments through simulation software. Heidergott and Vries [10] established a road network structure, used a maximum algebraic model, constructed a characteristic matrix with the delay of trains adjacent to the space of the late train as the eigenvalues, used stochastic optimal control methods to determine the model parameters, and analyzed whether a train was waiting to be connected. Delay the train, so as to monitor the delay of the entire road network. Sun et al. [11] established the GM(1,2) model to predict the train delay time, used the Markov matrix to correct the error, and used the radial neural network (RBF) to interpolate and expand the data to reduce the relative error. Bo et al. [12] used heuristic algorithms to obtain the operation plan after the train was disturbed and delayed under the conditions of the optimal speed and position when the train was disturbed, aiming to reduce the overall train delay in the research scope. Yuan [13] proposed novel multitrain trajectory optimization for single-track lines. They restrict attention to delay cases aimed at finding optimal speed profiles, which reduce delays and save energy consumption.

It can be seen that the current research on delay propagation theory and delay prediction mainly focuses on intelligent calculations. Traditional mathematical models are established to describe the order of train occupancy resources, so as to estimate the train arrival time or use a simulation system to simulate train operation. This method needs to be based on a certain prior scheduling knowledge and cannot achieve a completely objective and automatic delay propagation prediction. Under the current situation

of rapid development of artificial intelligence technology, it is possible to use machine learning and deep learning methods to establish data-driven train operation delay prediction models. However, there is no algorithm in the field of machine learning that can perfectly solve all problems, especially supervised learning. A single algorithm has various shortcomings, such as support vector machines being sensitive to missing data, neural networks falling into local minimums, and decision trees being prone to overfitting. Therefore, this paper uses the SSA method to smooth the train operation data and applies the idea of prediction fusion to the train delay prediction. Using the fusion method based on KNN, three different machine learning methods including the neural network, support vector machine, and random forest are fused to improve the prediction accuracy. The main contributions of this paper are summarized as follows:

- (i) We propose the idea of multimodel fusion to integrate different machine learning methods to improve the prediction accuracy of weak trainers
- (ii) We formulate a two-stage train delay prediction method based on data smoothing and multimodel fusion so that the prediction framework adds a data preprocessing stage and also uses the fusion result
- (iii) We use the KNN multimodel fusion method that the weight of each predictor is dynamically calculated and updated, so the unexpected traffic conditions can be easily incorporated into the fusion framework

The rest of the paper is organized as follows. Section 2 introduces the basic prediction model, including neural networks, SVM, and random forest. Section 3 introduces the predictive fusion framework. In Section 4, we propose the data smoothing method based on SSA and the multimodel fusion method based on KNN. Then, we introduce the process of experiment in Section 5 and present the numerical results in Section 6. Finally, we make a conclusion in Section 7.

2. Basic Prediction Model

In this paper, three commonly used machine learning methods, neural network, support vector machine, and random forest, are selected as individual predictors, and different algorithms are used to learn the relationship between two related data series to achieve the diversity of the model. The training process of the basic model is as follows: First, the model input is train operation data, including train speed, position, traction type, and other information, and the sample target value is the actual delay time. Secondly, through the adjustment and training of different algorithms, the model outputs satisfactory results. Finally, the train operation data is input to the model after the training is completed, and the model outputs the prediction of the train delay time.

2.1. Neural Networks. The neural network model is a data fitting model composed of a large number of nodes connected to each other. The complex relationship between multiple inputs and multiple outputs is captured through the weight configuration and activation function between nodes. As a data-driven method, neural networks do not need to specify explicit physical relationships between data in advance. The relationship between input and output is inferred from a training data set. The training process of the network model is to adjust the output value by repeatedly changing the node weights until the error between the output value and the sample target value is reduced to meet the requirements [14].

In this paper, the LSTM neural network model is selected for prediction. The model results are mainly input layer, hidden layer, and output layer, and the hidden layers all use multilayer LSTM units. The appropriate number of hidden layers and the appropriate number of neurons can be determined by the grid search method.

2.2. Support Vector Machines. Support vector machine is a two-class classification problem, which is a linear classifier with the largest interval defined in the feature space. Support vector machines can support linear classification of data as well as nonlinear data classification. The basic principle of its work is to find the optimal model with the largest interval. This paper uses the kernel function and soft interval maximization to learn nonlinear classifiers. The support vector machine maps the input space to the high-dimensional feature space through a certain nonlinear change $\varphi(x)$.

The dimensionality of the feature space can be very high. If the solution of the support vector machine only uses the inner product operation and there is a function $\kappa(x, x)$ in the low-dimensional input space, which is exactly equal to the inner product in the high-dimensional space, then the support vector machine does not need to calculate complex nonlinear transformations, and the inner product of the nonlinear transformation is directly obtained from this function $\kappa(x, x)$. Such a function $\kappa(x, x)$ is called a kernel function [15].

The choice of the kernel function requires the Mercer theorem to be satisfied; that is, any Gram matrix of the kernel function in the sample space is a positive semidefinite matrix. This paper selects the radial basis function kernel (RBF kernel):

$$k(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right), \quad (1)$$

where $r = \|x_1 - x_2\|$ and σ is a hyperparameter of the RBF core. This hyperparameter defines the feature length scale of the similarity between learning samples, that is, the ratio of the distance between the samples before and after the feature space mapping from the perspective of the weight space.

2.3. Random Forest. The random forest algorithm is a fusion algorithm based on decision tree classifiers proposed by Breiman. It belongs to the bagging algorithm in ensemble and is a classification and regression algorithm. The main

difference between the random forest and the traditional tree method is that the random forest randomly selects variables to divide each tree and node, which changes the tree construction method and avoids the problem of overfitting. The basic idea of this algorithm is to use the idea of the autonomous sampling method to randomly sample the original samples and construct several decision trees. When each decision tree is split, the optimal attribute is selected from some attributes for splitting. Finally, these decision trees are combined to form a random forest, and the average prediction of all decision trees in the random forest is used as the final prediction result. Random forest regression can be regarded as a strong predictor integrated with multiple weak predictors [16, 17].

There are two main factors that affect the performance of the random forest algorithm, which are the number of random forest algorithm trees and the number of optimal variables used in the binary tree in the specified node when building the tree. For random forests, the larger the number of trees and the larger the scale of the forest, the more stable the prediction results of the model. But at the same time, the more complex the model, the longer the running time. The optimal number of variables represents the accuracy of a single decision tree, so a reasonable optimal number of variables should be determined to minimize the value of the loss function.

3. Predictive Fusion Framework

The core concept of prediction fusion refers to the fusion of the results produced by different predictors. Because different prediction methods have different strengths and weaknesses under different conditions of normal and abnormal operations, instead of relying on a single prediction method, it is better to combine or fuse the results of different prediction methods in a similar way to sensor data fusion. Fusion is used in machine learning to combine individual learning processes. The main advantage of merging multiple methods is that they can take advantage of their complementary predictive properties. In order to combine multiple predictors, several different strategies can be used. The average method is the simplest way to combine multiple predictors. This method calculates the average of the predicted values of different individuals. A variant of the simple average method is the weighted method. In this method, each predictor has a weight, which is calculated based on the accuracy of the predictor on the validation data set. The final prediction is the sum of the weighted results of multiple predictors [18].

The basic idea behind the fusion-based framework is to generate the final prediction result by combining the outputs of two or more independent predictors. Figure 1 shows the prediction framework containing m ($m \geq 2$) predictors.

3.1. Average Fusion Method. This method uses the average of the results of multiple predictors as the final result. Given a set of predictors y^i , $i = 1, \dots, m$. The final prediction result is

$$\hat{y} = \frac{1}{m} (\hat{y}^1 + \hat{y}^2 + \dots + \hat{y}^m). \quad (2)$$

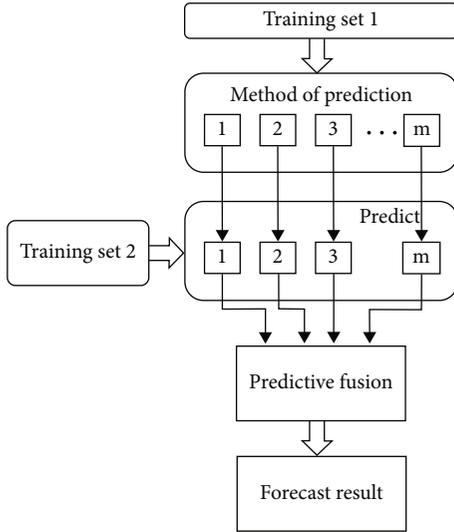


FIGURE 1: A framework for fusion-based prediction.

3.2. *Weighted Fusion Method.* In this method, different predictors have different weights. The weighted fusion method is denoted as

$$\hat{y} = \alpha_1 \hat{y}^1 + \alpha_2 \hat{y}^2 + \dots + \alpha_m \hat{y}^m = \sum_i \alpha_i \hat{y}^i \quad (3)$$

$$\alpha_1 + \alpha_2 + \dots + \alpha_m = 1,$$

where \hat{y}^i is the prediction result of the i th predictor and α_i is the weight of the predictor i . The weights are calculated from the training data set. When all weights are equal to $1/m$, the weighted fusion method is the same as the simple average method. This paper uses the inverse method of mean absolute percentage error (MAPE) to calculate the weights, $w = 1/\text{MAPE}$, $\hat{y} = \sum_i w_i \hat{y}^i / \sum_i w_i$.

The existing fusion methods all have the problem of low prediction accuracy, because the average fusion method does not consider the weights of the predictor, and although the weighted fusion method performs a weighted summation of the results of multiple predictors, the weight value is predetermined and constant during the prediction process. Therefore, these two methods are only suitable for train delay prediction under normal traffic conditions. Under abnormal traffic conditions caused by emergencies, equipment failures, and inclement weather, the established fusion model cannot provide accurate prediction results. In order to consider abnormal traffic conditions and asymmetry features in the train delay prediction model, this paper proposes a train delay prediction method based on KNN multimodel fusion. As a typical lazy learning method, the KNN method does not involve any explicit model construction and can locate the nearest observation value from the historical data set. The weight of each predictor is dynamically calculated and updated based on new observations, rather than predetermined. Taking into account the characteristics of KNN, unexpected traffic conditions can be easily incorporated into the fusion framework.

4. Two-Stage Train Delay Prediction Method Based on Data Smoothing and Multimodel Fusion

This paper designs a two-stage train delay prediction method based on data smoothing and multimodel fusion. In the two-stage framework, the first stage is to smooth train operation data and extract smooth components and residual components from noise data. The second stage uses different machine learning methods to train the smooth data and uses the KNN method to fuse different trainers. In addition, the historical average residual value is used as the prediction residual component, and finally, the prediction value of the smooth component and residual component is combined. Therefore, the framework essentially adds a data preprocessing stage to the common prediction model and also uses the fusion result of the smoothed residual value and multiple basic models.

The forecast framework is shown in Figure 2. In the application of train delay, suppose two types of data sequence input, historical (x_h) and current observation (x_c) train operation data. Historical operation data is used for training, and current observation data informs the current train operation status. In the first data smoothing step, the historical data x_h is decomposed into smooth x_{h-s} and its residual x_{h-r} in the offline process. At the same time, the residual sequence x_{h-r} is defined by the historical average value of the residual \hat{x}_{h-r} . In the online process, the x_{c-s} component in the observation data is smoothed by data smoothing extraction (and the residual part is discarded). The final prediction result is the sum of the predicted smoothing \hat{x}_{c-s} and the estimated residual \hat{x}_{h-r} based on historical data. Machine learning methods cannot effectively predict very irregular data patterns, such as residuals. Therefore, the framework only uses machine learning methods to predict the smooth sequence, and the residual part uses average analysis.

4.1. *Data Smoothing Method Based on SSA.* SSA is a data smoothing method used in the analysis of time series. It is widely used in many fields, such as hydrology and atmospheric and geophysical research. SSA is a model-free, adaptive noise-reduction algorithm based on the Karhunen-Loeve transform. It can also be used as a data denoising method by decomposing an original time series to a smoothed series and a noise series. Mineva and Popivanov [19] present a comprehensive description and discussion of the SSA method and identify a number of advantages of SSA compared to other data smoothing techniques. These advantages include the ability to characterize both trend and oscillatory components and the capability to reduce local noise and enhance pattern recognition and computational efficiency.

As a data denoising method, SSA can decompose the original time series into smooth components and residual components. The basic idea of the SSA method is to perform spectral analysis on the original input data to separate high-frequency “noise” components, thereby allowing the remaining components to be reconstructed into a smooth version of

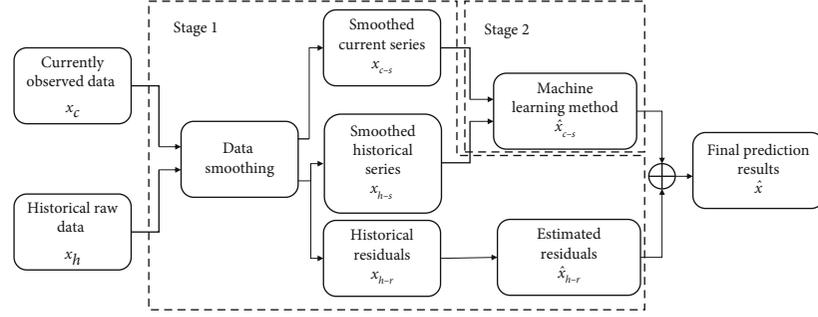


FIGURE 2: A framework of two-stage train delay prediction.

the original sequence. The SSA algorithm is usually implemented through the following four steps:

- (1) In this step, the original one-dimensional data X is converted into multidimensional data T_X , which can form a trajectory matrix
- (2) This step uses Singular Value Decomposition (SVD) to convert the trajectory matrix formed in step (1) into a decomposed trajectory matrix. The trajectory matrix T_X can be written as the following formula:

$$T_X = \sum_{i=1}^r u_i \sigma_i v_i^T. \quad (4)$$

Among them, u_i and v_i are the left and right eigenvectors of the trajectory matrix, and r is the rank of T_X . Element $u_i \sigma_i v_i^T$ is called the i th characteristic triplet of SVD.

- (3) This step reconstructs the decomposed trajectory matrix. This step is a grouping step, which corresponds to decomposing the matrix calculated in the SVD step into several groups and summing the matrices in each group. The extension of T_X can be written as

$$T_X = f_{\text{SVD}}(T_1, T_2, \dots, T_r) \quad (5)$$

- (4) Create a new time series through the grouping matrix in the third step. The corresponding operation is called diagonal averaging. It is a linear operation that maps the trajectory matrix of the initial series to the initial series itself. In this way, the initial series can be decomposed into several additive components

4.2. Multimodel Fusion Method Based on KNN. The KNN method is unstructured and does not require any predetermined model specifications. The basic idea of the KNN-based fusion method is to give the current train operating state, use the historical data set to search for the nearest neighbor to the state in training, calculate the prediction

error of the nearest neighbor, estimate the weight of each predictor, and combine the final prediction output of a single predictor based on these weights [20, 21]. Figure 3 describes the flow chart based on the KNN fusion method, which mainly includes two steps.

4.2.1. Proximity Search Process. The search process can find the nearest neighbor value, that is, the historical observation value that is most similar to the current observation value. There are two key design parameters for the KNN-based method: the distance metric and the choice of k . The distance metric is used to calculate the distance between two feature vectors. The distance metric may affect the result of data classification. However, in the time series prediction of large training data sets, the distance measurement is not the most important component. This paper uses Euclidean distance to determine the distance between the current input feature vector and historical observations.

Euclidean distance is the most common distance measurement method, which measures the absolute distance between two sample points in a multidimensional space. Assuming that the dimensions of the multidimensional space are N , $x_1 = x_{11}, x_{12}, \dots, x_{1N}$ and $x_2 = x_{21}, x_{22}, \dots, x_{2N}$ are two sample points, the formula for calculating the Euclidean distance between x_1 and x_2 is

$$L(x_1, x_2) = \sqrt{\sum_{n=1}^N (x_{1n} - x_{2n})^2}. \quad (6)$$

It can be seen that the Euclidean distance is calculated from the sum of the squares of the difference in each dimension between the two samples. When the value range between the dimensions is too different, the Euclidean distance is easily dominated by variables with a large value range, which will greatly reduce the effect of the model. Therefore, in the actual application of the K -nearest neighbor algorithm to solve problems such as classification, if the Euclidean distance is used as the similarity measure, it is best to standardize the data in advance.

k is the number of historical observation data closest to the input feature vector. Generally speaking, from the beginning $k=1$, with the gradual k increase, the classification effect of the K -nearest neighbor algorithm will gradually increase; after increasing to a certain value, as the further

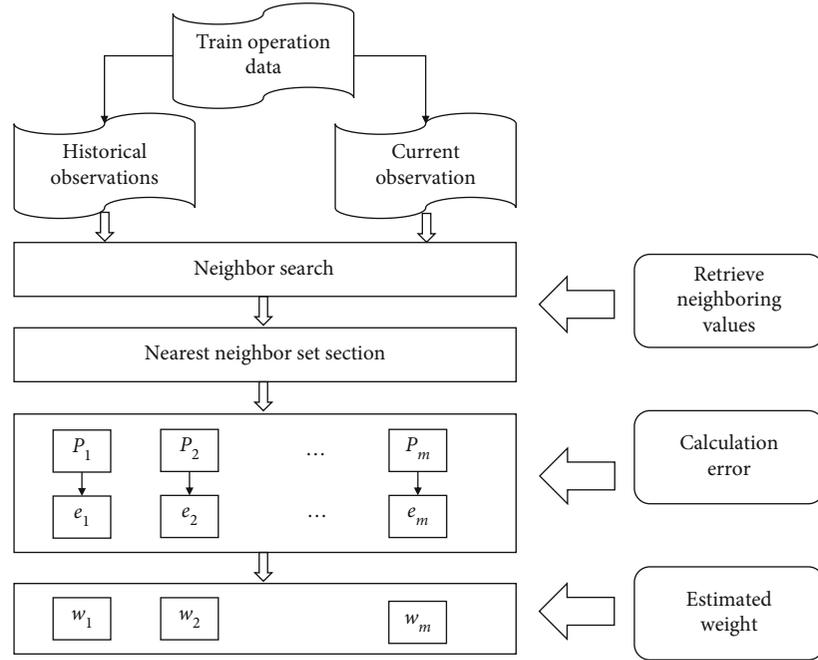


FIGURE 3: Flow chart of KNN fusion-based method.

increase, the classification effect of the K -nearest neighbor algorithm will gradually decrease. If the k value is too small, related neighboring values will be filtered out. If the k value is too large, noise will be introduced. The optimal value found in the research is related to the data and usually increases with the sample size and the variability of the data. In this paper, KNN parameters are set through cross-validation; that is, the sample data is divided into training data and verification data according to a certain ratio, starting from selecting a smaller k value, continuously increasing the value of k , and then calculating the variance of the verification set. And finally, find a more suitable k value. This article selects $k = 5$. The k -nearest neighbor set of the input feature vector x_c can be expressed as

$$J_{x_c} = [x_1, x_2, \dots, x_k]^T, \quad (7)$$

where $x_j = [x_1, x_2, \dots, x_n]$, $j = 1, 2, \dots, k$, n is the dimension of the feature space.

4.2.2. Weighted Parameter Estimation Process. This step is used to calculate the weight of each predictor. For each vector x_j , the predicted value $\hat{y}_j^i = f_i(x_j)$, $i = 1, 2, \dots, m$ and error $e_j^i = \hat{y}_j^i - y_i$ of the predictor can be calculated. Therefore, the error can be used to estimate the weight of each current predictor. $w_i = 1/\text{MAE}_i$ and MAE is calculated based on the selected nearest data set. The main difference between the KNN fusion method and the weighted fusion method is that the weights are dynamically updated at each step in the KNN fusion.

5. Experiment

The original data in this paper comes from actual rail transit lines, including actual train travel information tables, train schedules, driver shift tables, basic train information, and interlocation information. The details are shown in Table 1.

5.1. Data Preprocessing

5.1.1. Data Reading. First, the data of the same type in each file is converted into a unified format, and then, the data of a specific tag is used as a combination key for matching. This paper selects the three-column data combination of traffic date, train number, and location as the key and merges the data in each file.

5.1.2. Data Completion. First, we remove the duplicate data and use the key field as the deduplication key. Because some fields in the data are missing, missing value processing is required. There are three methods of missing value processing: directly use the feature with missing value, delete the feature with missing value, and complete the missing value. The processing method selected in this paper is the following: object type data includes abroad, pattern, traction type, location, and other missing data replaced by null values; float64 type data includes bare driving time, timetable speed, and other missing data are replaced by the median of this column, which represents universal condition.

5.1.3. Data Normalization. Due to the wide range of collected data, many feature dimensions and units are completely different, and the magnitude difference is obvious. It is necessary to eliminate the influence of different attributes of samples with different magnitudes. Data

TABLE 1: Data files and attributes.

File names	Descriptions	Properties
Realisation data	Actual train travel information table	Traffic date, train series, train characteristic, train number, location, activity, planner time, realisation
Timetable	Train schedule timetable	Day, train number, direction, location, activity, order number, planned time, pattern, train characteristic, bare driving time, traction type, timetable speed
Driver switches	Driver's shift schedule	Previous task, previous train, next task, next train, at location
Infor overview	Basic information	From location, to location, hectometers, average allowed speed, number of parallel tracks, diesel/ electric
Location distances	Interlocation information	From, to, distance

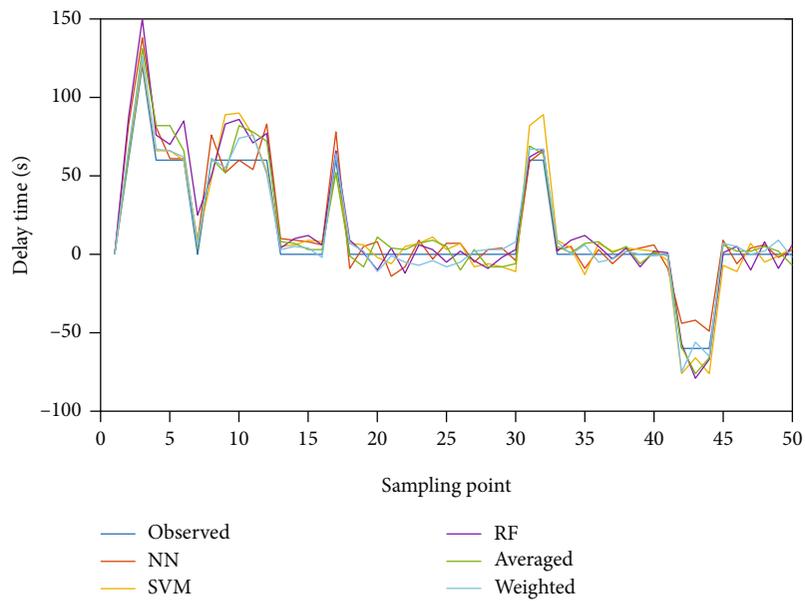


FIGURE 4: Comparison of real and predicted values of different methods in the training set.

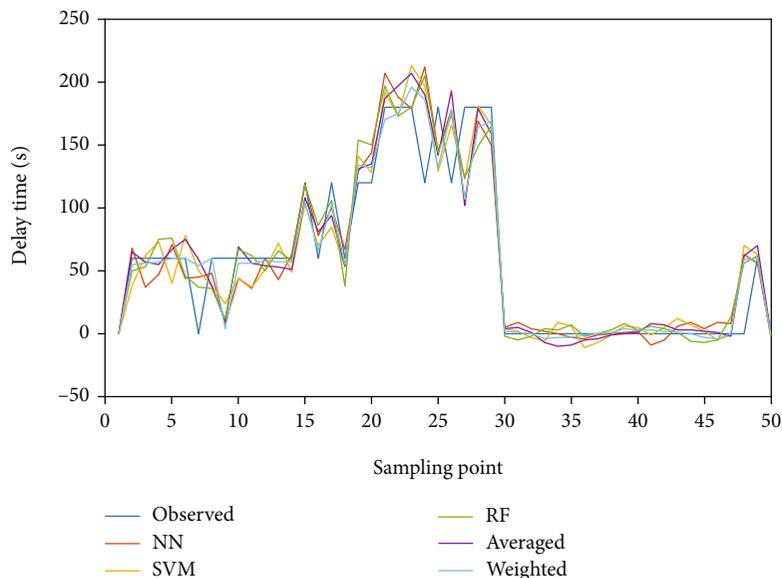


FIGURE 5: Comparison of real and predicted values of different methods in the test set.

TABLE 2: Predicted accuracy of different methods.

	Training set		Testing set	
	MAE	RSME	MAE	RSME
NN	7.80	6.29	16.08	25.86
SVM	9.02	11.78	16.34	24.46
RF	9.02	11.78	15.40	25.44
Baseline	8.61	9.92	15.94	25.25
Averaged	7.42	9.32	15.36	23.86
Weighted	6.74	8.63	15.22	23.84
KNN	5.02	6.24	12.08	23.14

normalization can speed up model training and improve model accuracy. This paper uses the z -score normalization method to process the data, so that the indicators are in the same order of magnitude. The calculation of the normalized data set needs to be based on the mean and standard deviation of the original data. The formula is as follows:

$$x' = \frac{x - \mu}{\sigma}. \quad (8)$$

In the formula, μ refers to the mean of all sample data; σ refers to the standard deviation of all sample data. After normalization, the data set conforms to the standard normal distribution; that is, the standard deviation is 1 and the mean is 0.

5.2. Feature Selection. The process of selecting relevant feature subsets from a given feature set is called feature selection. The two main reasons for feature selection are to reduce the dimensionality disaster problem and to reduce the difficulty of the learning task. Feature selection must ensure that important features are not lost. Common feature selection methods are divided into three categories: filtering, wrapping, and embedded.

The filtering method scores each feature according to divergence or relevance and sets a threshold or the number of thresholds to be selected to select features. The wrapping method selects several features or excludes several features each time according to the objective function, until the best subset is selected. This method directly uses the performance of the learner to be used as the evaluation principle of the feature subset. Its advantage is that it is directly optimized for a specific learner, but the disadvantage is that the computational cost of training the learner multiple times is large. The embedding method first uses certain machine learning algorithms and models for training, obtains the weight coefficient of each feature, and selects the feature from large to small according to the coefficient. In this paper, the PCA method is used for feature selection.

PCA, the principal component analysis method, is one of the most widely used data dimensionality reduction algorithms. The main idea of PCA is to map n -dimensional features to k -dimensions. This k -dimension is a brand new orthogonal feature, also called principal component, which is a k -dimensional feature reconstructed on the basis of the

original n -dimensional feature. The job of PCA is to sequentially find a set of mutually orthogonal coordinate axes from the original space. The choice of new coordinate axes is closely related to the data itself. Among them, the first new coordinate axis selection is the direction with the largest variance in the original data, the second new coordinate axis selection is the plane orthogonal to the first coordinate axis that maximizes the variance, and the third axis has the largest variance in the plane orthogonal to the first and second axes. By analogy, n such coordinate axes can be obtained. With the new coordinate axis obtained in this way, we find that most of the variance is contained in the first k coordinate axes, and the variance contained in the latter coordinate axis is almost zero. Therefore, we can ignore the remaining coordinate axes and keep only the first k coordinate axes that contain most of the variance. In fact, this is equivalent to only retaining the dimensional features that contain most of the variance and ignoring the feature dimensions that contain almost zero variance, so as to realise the dimensionality reduction processing of the data features. The specific implementation method is as follows:

Input: data set $X = \{x_1, x_2, \dots, x_n\}$, which needs to be reduced to k dimensions.

- (1) Deaveraging (decentralization), that is, each feature minus its average
- (2) Calculate the covariance matrix $(1/n)XX^T$
- (3) Calculate the eigenvalues and eigenvectors of the covariance matrix $(1/n)XX^T$ through SVD
- (4) Sort the eigenvalues from large to small and select the largest k among them. Then, the corresponding k eigenvectors are used as row vectors to form the eigenvector matrix P
- (5) Convert the data into a new space constructed by k feature vectors, that is, $Y = PX$

5.3. Forecast Accuracy Measurement. In this paper, mean absolute error (MAE) and root mean square error (RMSE) are used to evaluate the final prediction accuracy. MAE is the average of the absolute values of the deviations of all single true values and predicted values. MAE can avoid the problem of mutual cancellation of errors, so it can accurately reflect the size of the actual prediction error. The mean square error (MSE) is the square of the difference between the true value and the predicted value, and then, the sum is averaged. RMSW is a square root based on MSE, which can provide additional weight for larger absolute errors. The definitions of these two methods are as follows:

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|, \quad (9)$$

$$\text{RMSE} = \text{sqrt} \left[\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \right]. \quad (10)$$

Among them, m is the number of predicted results, \hat{y}_i is the true value, and y_i is the predicted value.

6. Experimental Results

The prediction of train delays uses independent machine learning methods such as neural networks, support vector machines, and random forests and uses the average fusion method, weighted fusion method, and KNN-based fusion framework. Figures 4 and 5 are the delay time diagrams of the prediction results using different methods in the training set and test set, respectively. Table 2 compares the prediction accuracy of different methods. Among them, the average of the MAE and RMSE of the three independent methods is used as the baseline to test the accuracy improvement based on the fusion method.

In the training set, using three independent machine learning methods to predict train delays, the average MAE value is 8.61, and the RSME value is 9.92. Based on average fusion, weighted fusion, and KNN fusion, the MAE values were 7.42, 6.74, and 5.02, and the RSME values were 9.32, 8.63, and 6.27. In the test set, using three independent machine learning methods to predict train delays, the average MAE value is 15.94, and the RSME value is 25.25. The MAE values based on average fusion and weighted fusion were 15.36, 15.22, and 12.08, and RSME values were 23.86, 23.84, and 23.14. It can be seen that the accuracy of the three different machine learning methods for predicting train delays is not much different, and the prediction accuracy of the fusion framework is greater than that of the independent method, and the fusion method based on KNN is better than the weighted fusion and average fusion accuracy.

7. Conclusions

Urban rail transit trains are easily affected by factors such as passenger flow, dispatch and command, equipment failure, traffic accidents, and extreme weather. The train operation features under these abnormal traffic conditions are asymmetry, and the train delay prediction method through the establishment of traditional mathematical models is uncertain. This paper uses machine learning methods to propose a framework that integrates multiple predictors to reduce these uncertainties and improve the performance of train delay prediction. In this paper, a series of experiments are carried out using actual subway train operation data at different times to evaluate the accuracy of different machine learning methods and different fusion methods. The experimental conclusions are as follows:

- (1) The effect of supervised learning using a single algorithm cannot fully meet specific needs. For example, the experimental results show that the prediction effects of the three independent prediction models are equivalent, but none of them can achieve satisfactory accuracy
- (2) Using the complementary advantages of different machine learning algorithms to fuse multiple inde-

pendent predictors can improve the final prediction results. For example, the experimental results show that the train delay prediction accuracy of the three fusion models is higher than that of the independent predictor

- (3) The fusion method is very important to the performance of the fusion model. As the experimental results show, the performance of the KNN-based multimodel fusion method is better than the average fusion method and the weighted fusion method

Data Availability

The original data in this paper comes from actual rail transit lines, including actual train travel information tables, train schedules, driver shift tables, basic train information, and interlocation information.

Conflicts of Interest

The author declares that they have no conflicts of interest.

References

- [1] J. W. Sun, A. Zhang, and Z. X. Zhou, "Research on urban rail transit promoting the process of urbanization," *China Soft Science*, vol. 6, pp. 96–111, 2020.
- [2] Y. Li, L. Zhu, H. Wang, F. R. Yu, and S. Liu, "A cross-layer defense scheme for edge intelligence-enabled CBTC systems against MitM attacks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2286–2298, 2021.
- [3] L. Zhu, Y. Li, F. R. Yu, B. Ning, T. Tang, and X. Wang, "Cross-layer defense methods for jamming-resistant CBTC systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7266–7278, 2020.
- [4] H. Liang, Y. Zhang, and H. Xiong, "A blockchain-based model sharing and calculation method for urban rail intelligent driving systems," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–5, Rhodes, Greece, September 2020.
- [5] L. Zhu, H. Liang, H. Wang, B. Ning, and T. Tang, "Joint security and train control design in blockchain empowered CBTC system," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8119–8129, 2022.
- [6] H. Liang, H. Zhao, S. Wang, and Y. Zhang, "LTE-U based train to train communication system in CBTC: system design and reliability analysis," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8893631, 14 pages, 2020.
- [7] Y. Zhao, "Research on the self-organization mechanism of metro operation adjustment under the condition of train delays," *Urban Rail Transit Research*, vol. 24, no. 4, pp. 13–17, 2021.
- [8] G. Qing, J. Li, and Z. Zhang, "Research on automatic recovery strategy of metro train operation delay," *Control and Information Technology*, vol. 4, pp. 49–53, 2020.
- [9] S. Harrod, F. Cerreto, and O. A. Nielsen, "A closed form railway line delay propagation model," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 189–209, 2019.

- [10] B. Heidergott and R. D. Vries, "Towards a (max,+) control theory for public transportation networks," *Discrete Event Dynamic Systems*, vol. 11, no. 4, pp. 371–398, 2001.
- [11] L. Sun, R. S. Song, and H. Shi, "Prediction method of train delay time in technical station," *Journal of Beijing Jiaotong University*, vol. 42, no. 1, pp. 94–98, 2018.
- [12] Y. Bo, T. He, and B. Mao, "Train control to reduce delays upon service disturbances at railway junctions," *Transportation System Engineering and Information*, vol. 11, no. 5, pp. 114–122, 2011.
- [13] Z. Yuan, "Optimization strategy and method of train delay control on complicated line," *China Academy of Railway Sciences*, vol. 105, pp. 340–361, 2017.
- [14] C. Wen, Z. Li, and P. Huang, "Research on data-driven train delay propagation," *Chinese Safety Science Journal*, vol. 29, no. S2, pp. 1–9, 2019.
- [15] Q. Zhang, F. Chen, and T. Zhang, "Intelligent prediction and feature recognition of associated delays of high-speed railway trains," *Acta Automatica Sinica*, vol. 45, no. 12, pp. 2251–2259, 2019.
- [16] A. Verikas, M. B. Gelzinis, and M. Bacauskiene, "Mining data with random forests: a survey and results of new tests," *Pattern Recognition*, vol. 44, no. 2, pp. 330–349, 2011.
- [17] W. Mou, "Research on the prediction of train operation delay in Dutch railway network based on data-driven," *Southwest Jiaotong University*, vol. 11, pp. 54–64, 2018.
- [18] F. Guo, J. W. Polak, and R. Krishnan, "Predictor fusion for short-term traffic forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 90–100, 2018.
- [19] A. Mineva and D. Popivanov, "Method for single-trial readiness potential identification, based on singular spectrum analysis," *Journal of Neuroscience Methods*, vol. 68, no. 1, pp. 91–99, 1996.
- [20] F. Guo, R. Krishnan, and J. Polak, "The influence of alternative data smoothing prediction techniques on the performance of a two-stage short-term urban travel time prediction framework," *Journal of Intelligent Transportation Systems*, vol. 21, no. 3, pp. 214–226, 2017.
- [21] H. Lan and M. Huang, "Fusion of KNN optimized density peak and FCM clustering algorithm," *Computer Engineering and Applications*, vol. 57, no. 9, pp. 81–88, 2021.